

AMENDMENTS TO CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A communication process for enabling a server end to communicate with a system under verification (SUV) in a network by connecting ~~a the~~ server end in series with ~~a system under verification (SUV) in a network~~, the communication process comprising the steps of:

(a) initializing a communication port connected to the server end and the SUV through an initialization module in each of the server end and the SUV by a computer in one of the server end and the SUV;

(b) processing data to be sent prior to sending to a predetermined buffer in a data transmission module in one of the server end and the SUV;

(c) sending data to a data receiving module of the other connected one of the server end and the SUV through the predetermined buffer in the data transmission module in one of the server end and the SUV;

(d) storing the received data in a predetermined buffer in the data receiving module in the other one of the server end and the SUV ~~prior to and only~~ sending the data from the data receiving module to a cleaning module when a complete data package has been stored;

(e) ~~deleting causing the cleaning module to delete~~ an associated head contained in the data ~~by the cleaning module~~ so as to obtain the original data sent therefrom; and

(f) continuing to perform the steps (a) - (e) for transmitting data between the server end and the SUV through the connected communication port.

2. (Currently Amended) The communication process of claim 1, wherein in performing the step of initialization by through the initialization module in the server end in step (a), the process

further ~~comprising~~ comprises the steps of commanding the computer to assign a data storage buffer in the server end to the data transmission module and the data receiving module respectively, determining whether the assignment ~~is~~ has succeeded[[,]] such that if the assignment fails, aborting the process ~~aborts~~[[,]] and otherwise assigning the communication port ~~is assigned~~ to the computer based on an embedded communication port parameter, initializing the communication port and a storage associated with the communication port, creating a thread, and determining whether the initialization and thread creation ~~are~~ have succeeded, wherein if the initialization and thread creation have succeeded, the process ends normally ~~otherwise and if the~~ the initialization and thread creation have not succeeded, the process aborts.

3. (Currently Amended) The communication process of claim 1, wherein in performing the step of initialization by through the initialization module in the server end in step (a), the process further ~~comprising~~ comprises the steps of commanding the computer to assign a data storage buffer in the server end to the data transmission module and the data receiving module respectively, determining whether the assignment ~~is~~ has succeeded[[,]] such that if the assignment fails, aborting the process ~~aborts~~[[,]] and otherwise assigning the communication port ~~is assigned~~ to the computer based on an embedded communication port parameter, initializing the communication port and a storage associated with the communication port, creating an interrupt program based on ~~the~~ a thread in the server end, and determining whether the initialization and ~~the~~ a thread creation ~~are~~ have succeeded, wherein if the initialization and thread creation have succeeded, the process ends normally ~~otherwise and if the~~ the initialization and thread creation have not succeeded, the process aborts.

4. (Currently Amended) The communication process of claim 3, wherein in processing the thread and the interrupt program the computer in each of the server end and the SUV, the process further ~~comprising~~ comprises the steps of continuously monitoring the status of the communication port connected to the server end and the SUV for determining whether data has been transmitted to the communication port, if no data has been received, returning to the monitoring step[[,]] and if ~~yes~~ data has been received, searching a complete data package in the predetermined buffer in

the data receiving module, if there is a complete data package, the process returns to the monitoring step, if there is no complete data package, the process receives data based on the head contained in the data package, determining whether ~~the data package is complete~~ a set value is contained in the data package, if not, returning to the monitoring step, if yes, retrieving a data size bit in the data package for determining ~~a whether the~~ whether the set value of the data package ~~to be is~~ equal to 0, if ~~yes~~ the set value is equal to 0, ~~returning~~ to the monitoring step, if the set value is not equal to zero, receiving data based on the data size bit of the data package, and performing a processing on the data based on a data type thereof.

5. (Currently Amended) The communication process of claim 1, wherein in the step (c) of sending data to a data receiving module of the other connected one of the server end and the SUV, the process further ~~comprising~~ comprises the steps of commanding the computer in one of the server end and the SUV to issue a transmission request to the other connected one of the server end and the SUV and waiting for a reply, determining whether there is a reply, if there is a reply, commanding the computer to receive the request and transmit the message contained in the request to the other connected one of the server end and the SUV for receiving, determining whether the request is accepted by the receiving end, if not aborting the process, if ~~yes~~ the request is accepted, commanding the computer to transmit data, the computer determining whether there is a reply from the receiving end simultaneously, if ~~yes~~ there is a reply from the receiving end simultaneously, commanding the computer to determine whether the transmission has ended, if ~~not the transmission has not ended~~, the process looping back to the data transmission step, if the transmission has ended, an end of a transmission flag is sent to the receiving end, and determining whether there is a reply from the receiving end with respect to the end of the transmission flag, if ~~yes~~ there is a reply from the receiving end with respect to the end of the transmission flag, ending the process ~~ends~~ normally.

6. (Currently Amended) The communication process of claim 5, wherein in commanding the computer in one of the server end and the SuV to issue a transmission request to the other connected one of the server end and the SUV and waiting for a reply, if there is no reply the

process further ~~comprising~~ comprises the steps of determining whether the waiting is within a predetermined limit, if the waiting is within the predetermined limit, the process loops back to the waiting state, if not determining whether the time of requesting transmission ~~has~~ have reached a predetermined value, if yes, the process loops back to the transmission request step, if not, the process aborts.

7. (Currently Amended) The communication process of claim 5, wherein in commanding the computer to transmit data, if the computer determines there is no reply from the receiving end the process further ~~comprising~~ comprises the steps of determining whether the waiting is within a predetermined limit, if the waiting is within the predetermined limit, the process loops back to the waiting state, if not, determining whether the times of requesting transmitting data ~~has~~ have reached a predetermined value, if yes, the process loops back to the data transmission step, if not, the process aborts.

8. (Currently Amended) The communication process of claim 5, wherein in sending the end of the transmission flag the process further ~~comprising~~ comprises the steps of determining whether the waiting is within a predetermined limit, if the waiting is within the predetermined limit, the process loops back to the waiting state, if not, determining whether the times of requesting sending the end the transmission flag ~~has~~ have reached a predetermined value, if yes, the process loops back to the step of requesting sending of the end of the transmission flag ~~step~~, if not, the process aborts.

9. (Currently Amended) The communication process of claim 1, wherein when data sent from the other connected one of the server end and the SUV is received by one of the server end and the SUV, the process further ~~comprising~~ comprises the steps of the computer in ~~on one~~ one of the server and the SUV ~~determines~~ determining whether the received data is in compliance with the data type contained in the data package, if not, the process ends, if yes, determining whether a predetermined user buffer is full, if yes, the process ends, if not, the process writes the received data from the predetermined buffer in the data receiving module into the predetermined user

Serial Number 09/726,319

buffer, and determining whether the end of transmission has been received, if yes, the process ends ~~otherwise~~ and if not, the process loops back to the step of determining whether the received data is in compliance with the data type contained in the data package.